MANUAL DE USUARIO

EXPANDER PI

MCI-MA-0229 | REV. 1.0

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 of. 1105, Providencia, Santiago, Chile.

+56 2 23339579 | www.olimex.cl | info@olimex.cl

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 Oficina 1105
Providencia, Santiago, Chile

www.olimex.cl
info@olimex.cl

Tel: +56 2 23339579
Fax: +56 2 23350589

® MCI Ltda. 2014

Atención: *cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía.*

Código Manual: MCI–MA-0229

CONTENIDO

Contenido 3
INTRODUCCIÓN 5
PARTES DEL DISPOSITIVO 5
ACERCA DE LA LIBRERÍA 6
1.1 INSTALACIÓN DE LA LIBRERÍA 6
1.2 CONFIGURACIÓN DEL BUS I2C 7
1.3 COMPILACIÓN DE PROGRAMAS 7
DOCUMENTACIÓN DE LA LIBRERÍA 7
2.1 DIRECCIONES I²C 7
2.2 MAPEO DE PINES 8
2.3 INICIALIZACIÓN DE LA LIBRERÍA 10
2.4 GPIO 10
2.4.1 Configuración del integrado: mcp23016Setup() 10
2.4.2 Configuración de un pin: pinMode() 10
2.4.3 Escritura digital: digitalWrite() 11
2.4.4 Lectura digital: digitalRead() 11
2.5 ADC 11
2.5.1 Configuración del integrado: mcp3424Setup() 11
2.5.2 Lectura análoga: analogRead () 12
2.6 PWM 12
2.6.1 Configuración del integrado: pca9685Setup() 13
2.6.2 Escritura analógica (PWM): pwmWrite() 13
Ejemplos 15
3.1 Entrada GPIO 15
3.2 SALIDA GPIO 16

3.3 ADC 17

3.4 PWM 18

SOLUCIÓN DE PROBLEMAS 20

4.1 La Expander Pi no realiza algunos comandos que le son enviados 20

4.2 El ejemplo de salida GPIO con el relé funciona de forma errática. 20

4.3 Error de compilación:fatal error: wiringPi.h: No such file or directory 20

4.4 Error de enlazado: undefined reference to `wiringPiSetup' 21

CARACTERÍSTICAS ELÉCTRICAS 21

CARACTERÍSTICAS MECÁNICAS 21

HISTORIA DEL DOCUMENTO 21

INTRODUCCIÓN

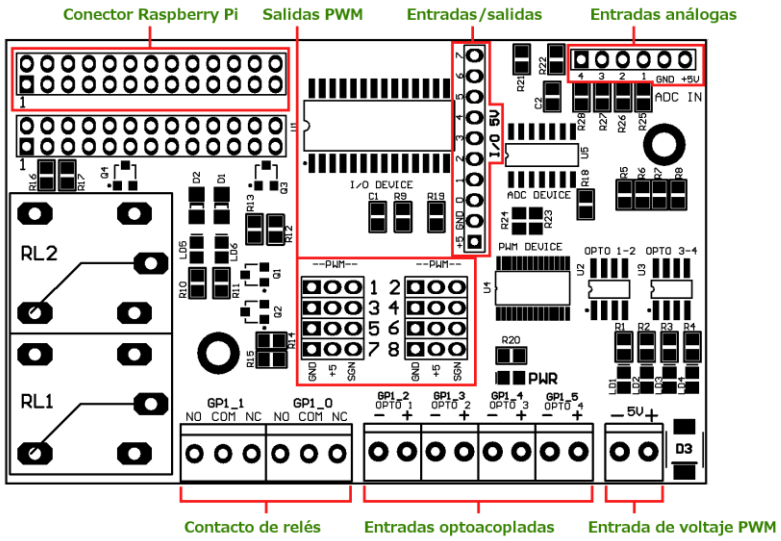
La placa Raspberry Pi al ser un verdadero computador con microprocesador, carece de algunas funcionalidades clásicas del mundo de los microcontroladores. Mediante el bus I²C, la Expander Pi permite extender las capacidades agregando pines I/O, conversores análogo digital (ADC) y modulación por ancho de pulso (PWM). La placa también cuenta con relés y optoacopladores listos para ser usados en todos sus proyectos de electrónica.

Para hacer uso de los pines de expansión presentes en la placa, se hace uso de la librería wiringPi similar a Wiring de Arduino.

Este manual incluye los pasos de instalación de la librería, la descripción de las funciones que permiten controlar los distintos pines, ejemplos de apoyo y soluciones a los problemas más frecuentes.

PARTES DEL DISPOSITIVO

A continuación se presentan las partes más relevantes de la tarjeta.



- Conector Rasberry Pi: pines para unir la Raspberry Pi con esta tarjeta
- Salidas PWM: pines de salida PWM con pines de alimentación externa
- Entradas/ salidas: pines configurables como entradas o salidas tolerante a 5V
- Entradas análogas: pines para agregar sensores análogos
- Contacto relés: terminal block con los contactos correspondiente a los relés
- Entradas optoacopladas: entradas que soportan hasta 24VDC
- Entrada de voltaje PWM: para alimentar de forma externa el conector del PWM

ACERCA DE LA LIBRERÍA

La librería que permite usar los pines presentes en la Expander Pi, está basada en wiringPi, una librería que controla los pines del SoC BCM2835 presentes en la Raspberry Pi. Fue escrita en C por Gordon Henderson, liberada bajo la licencia GNU LGPLv3 y se puede usar en C, C++ y mediante adaptadores (wrappers) en muchos otros lenguajes. WiringPi es similar en el nombre de las funciones a*Wiring*de Arduino y puede ser extendida para controlar chips de otros circuitos integrados que estén presentes en el bus I²C.Para mayor información, visitar el sitio http://wiringpi.com/

1.1 INSTALACIÓN DE LA LIBRERÍA

Si usted tiene el sistema operativo Raspbian, conecte la Raspberry Pi a Internet y ejecute los siguientes comandos en la shell (línea de comandos) y conteste a las preguntas que aparecerán en pantalla:

cd /home/pi

wget http://www.olimex.cl/documents/manuals/instalador_expanderPi.zip

tar -zxf instalador_expanderPi.zip

cd instalador_expanderPi

chmod +x instalador_expanderPi.sh

./instalador_expanderPi.sh

Si tiene otro sistema operativo, instale el software I2C-TOOLS desde su gestor de paquetes o bien desde el código fuente presente en el sitiohttp://www.lm-sensors.org/wiki/I2CTools, y luego ejecute los siguientes comandos:

cd /home/pi

wget http://www.olimex.cl/documents/manuals/expanderPi.tar.gz

tar -zxf expanderPi.tar.gz

cd expanderPi

tar -zxf wiringPi-modified.tar.gz

cd wiringPi-modified

./build

Finalmente, agregue **gpio load i2c 50** a su cargador de scripts de arranque para que el módulo controlador de I²C sea cargado al bootear.

1.2 CONFIGURACIÓN DEL BUS I2C

Para poder hacer uso de la placa Expander Pi con su Raspberry Pi, debe cargar el módulo controlador i2c_bcm2708. Para ello ejecute:

gpio load i2c 50

en donde el número 50 al final del comando indica la velocidad de transmisión (baudrate) a la que funcionará el bus I²C, que en este caso está funcionando a 50 Kb/s. Modifique este valor según necesidad para aumentar la inmunidad frente al ruido en la comunicación entre las placas, o para aumentar la velocidad.

La configuración del bus deberá realizarla cada vez que reinicie el sistema, a menos que haya creado, o bien, haya permitido al instalador de la librería crear el script /etc/init.d/expanderPi, para cargar el controlador al bootear el sistema operativo.

1.3 COMPILACIÓN DE PROGRAMAS

Si está trabajando con un IDE, y este no enlaza automáticamente la librería, deberá hacerlo en forma manual apuntando al archivo /usR/LOCAL/LIB/LIBWIRINGPI.SO. El proceso de enlazar una librería depende de cada IDE, por lo que debe dirigirse a la documentación de éste para mayor información. Para compilar y enlazar desde la línea de comando, ejecute:

gcc codigo.c -lwiringPi -o ejecutable

donde CODIGO.C es el archivo que contiene el código fuente, y EJECUTABLE es el nombre del archivo ejecutable que se generará.

DOCUMENTACIÓN DE LA LIBRERÍA

En esta sección se presenta la documentación de la librería. Para poder hacer uso de los pines presentes en la placa, usted debe conocer cuáles son las direcciones I²C de los circuitos integrados presentes en la placa y el número del pin lógico que quiere controlar.

2.1 DIRECCIONES I²C

La Expander Pi tiene tres circuitos integrados que están conectados al bus I²C, cuyas direcciones I²C y funciones desempeñadas son mostradas en la Tabla.

Dispositivo	Dirección	Función
mcp23016	26	Pines GPIO

mcp3424	6E	Pines ADC
pca9685	5C	Pines

Tabla 1: Dirección I²C y función de los circuitos integrados presentes en la Expander Pi.

Para comprobar las direcciones desde la shell, si está usando una Raspberry Pi versión.2 (rev.2), ejecute:

sudo i2cdetect 1

Y si está usando una Raspberry Pi versión 1 (rev.1), escriba:

sudo i2cdetect 0

Como resultado verá en pantalla algo similar a la

Ilustración 1, en donde como mínimo debe ver los números 26, 5c y 6e, lo cual le indicará que existe comunicación entre las dos placas y los tres circuitos integrados están funcionando. No tome en cuenta el número 70.

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- 26 -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- 5c -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 6e --
70: 70 -- -- -- -- -- -- --
```

Ilustración 1: Obteniendo las direcciones I²C

2.2 MAPEO DE PINES

En wiringPi, para acceder a los pines GPIO de la Raspberry Pi y a pines de expansión de otros integrados compatibles, se debe realizar un mapeo de pines. Este mapeo, es una asociación de pines lógicos a pines físicos, mediante el cual se permite el control y el uso del hardware.

Para el mapeo, existen pines lógicos estáticos y dinámicos. Los estáticos hacen referencia a los pines que están presentes en el integrado BCM2835, perteneciente a la Raspberry Pi. Estos, están fijos, por lo que no se pueden modificar y van desde el 0 hasta el 63. Los pines dinámicos, son útiles para hardware externo a la Raspberry Pi, como en la Expander Pi. Estos pines no están fijos y

dependen de un valor llamado<PINBASE> que usted debe elegir para cada uno de los integrados. El <PINBASE> es el número base desde donde comienza el mapeo de pines lógicos de la librería a los pines físicos del hardware externo. Es por esto que el <PINBASE> debe ser un número mayor a 64 para que no exista un conflicto con los números de pines de la Raspberry Pi y escogido con el debido cuidado para que tampoco entre en conflicto con pines de otro integrado.

Para ilustrar mejor la situación, supongamos que se eligen los números de <PINBASE> para cada uno de los circuitos integrados como se muestra en la Tabla 2.

pinBase	Circuito
100	MCP23016
200	MCP3424
300	PCA9685

Tabla 2: Ejemplo de pinBase para los distintos circuitos integrados

De este modo, se obtendrá el mapeo mostrado en la Tabla 3.

Pin	Descripción	Pin	Descripción
0	Raspberry Pi	**199**	No asignado
1	Raspberry Pi	**200**	Canal 0 del chip MCP3424
…	Raspberry Pi	**201**	Canal 1 del chip MCP3424
62	Raspberry Pi	**202**	Canal 2 del chip MCP3424
63	Raspberry Pi	**203**	Canal 3 del chip MCP3424
64	No asignado	**204**	No asignado
65	No asignado	**205**	No asignado
…	No asignado	**…**	No asignado
98	No asignado	**298**	No asignado
99	No asignado	**299**	No asignado
100	Pin 1 del chip MCP23016	**300**	PWM 1 del chip PCA9685
101	Pin 2 del chip MCP23016	**301**	PWM 2 del chip PCA9685
…	Pin n del chip MCP23016	**…**	PWM n del chip PCA9685
114	Pin 15 del chip MCP23016	**314**	PWM 15 del chip PCA9685
115	Pin 16 del chip MCP23016	**315**	PWM 16 del chip PCA9685
116	No asignado	**316**	No asignado
117	No asignado	**317**	No asignado
…	No asignado	**…**	No asignado
198	No asignado	**…**	No asignado

Tabla 3: Ejemplo de mapeo de pines

De acuerdo a las tablas, es posible notar que no existe conflicto entre los pines de la Raspberry Pi ni los de los distintos integrados de la placa Expander Pi. Según el mapeo anterior, si se deseara realizar una lectura análoga sobre el canal 2 del chip MCP3424, se debe utilizar el pin lógico número 202. De igual modo, para utilizar el pin 15 del chip MCP23016, se debe utilizar el pin lógico número 114.

2.3 INICIALIZACIÓN DE LA LIBRERÍA

Debe incluir el archivo *wiringPi.h*e inicializar la librería ejecutando la función wiringPiSetup() sin argumentos. Si necesita mayor información, puede obtenerla en la página oficial de la librería wiringPi: https://projects.drogon.net/raspberry-pi/wiringpi/functions/.

2.4 GPIO

La Expander Pi cuenta con 16 pines digitales para utilizar en sus proyectos. Para inicializarlos, debe realizar la configuración del integrado MCP23016 llamando a la función mcp23016Setup(). Luego podrá escoger entre los modos de entrada o salida para cada pin con pinMode() y realizar escrituras con digitalWrite(), o bien, lecturas con digitalRead().

2.4.1 CONFIGURACIÓN DEL INTEGRADO: MCP23016SETUP()

int mcp23016Setup (const int pinBase, const int i2cAddress);

Carga la configuración inicial del chip MCP23016, estableciendo el número base de los pines <PINBASE> y la dirección I²C*<i2cAddress>* del chip.

- <PINBASE>: número base de los pines de este chip. <PINBASE> debe ser un número mayor que 64, tal que el rango de pines lógicos [pinBase, pinBase + 16] no esté siendo utilizado por otro circuito integrado.

<I2CADDRESS>: dirección del chip MCP3424. La dirección puede ser consultada en la sección 0 “

- ***DIRECCIONES I²C***”.

Retorna cero si no hubo error o un valor distinto de cero en caso contrario.

3.1.1 CONFIGURACIÓN DE UN PIN: PINMODE()

void pinMode (int pin, int mode);

Configura el modo *<mode>* de funcionamiento del pin lógico *<pin>.*

- <PIN>: es el número de pin lógico sobre el cual se configurará su modo.

- *<mode>:* es el modo de funcionamiento que se establecerá sobre el pin. Puede ser *OUTPUT* para salida o *INPUT* para entrada.

No retorna valor.

3.1.2 ESCRITURA DIGITAL: DIGITALWRITE()

void digitalWrite (int pin, int value);

Establece el estado *<value>*en el pin lógico*<pin>.*

- *<pin>:* es el número de pin lógico sobre el cual se establecerá el estado*<value>.*
- *<value>:* estado lógico. Puede ser *HIGH* o *LOW*;

No retorna valor.

3.1.3 LECTURA DIGITAL: DIGITALREAD()

int digitalRead (int pin);

Lee el estado del pin lógico*<pin>*

- *<pin>:* es el número de pin lógico sobre el cual se leerá el estado.

Retorna *HIGH* o *LOW*.

3.2 ADC

La Expander Pi cuenta con 4 pines análogos para utilizar en sus proyectos. Para inicializarlos debe realizar la configuración del integrado MCP3424 llamando a la función mcp3424Setup(), en la cual, debe establecer la tasa de muestreo y la ganancia. Luego podrá realizar lecturas análogas utilizando la función analogRead().

3.2.1 CONFIGURACIÓN DEL INTEGRADO: MCP3424SETUP()

int mcp3424Setup (int pinBase, int i2cAddress, int sampleRate, int gain);

Carga la configuración inicial del chip MCP3424, estableciendo el número base de los pines *<pinBase>,* la dirección I²C*<i2cAddress>,* la frecuencia de muestreo *<sampleRate>* y la ganancia *<gain>.*

- *<pinBase>:* número base de los pines de este chip. <pinBase> debe ser un número mayor que 64, tal que el rango [pinBase, pinBase + 3] no esté siendo utilizado por otro chip de la librería wiringPi.

<i2cAddress>: dirección del chip MCP3424. La dirección puede ser consultada en la sección 0 “

- ***DIRECCIONES I²C***".
- *<sampleRate>:* indica la frecuencia de muestreo con la que funcionará el ADC. Los valores posibles se muestran en la Tabla 4.
- *<gain>:* indica la ganancia que se aplicará a la señal antes de realizar la conversión análoga a digital, permitiendo convertir señales débiles con una alta resolución. Los posibles valores se muestran en la Tabla 5.

Retorno: 0 si no hubo error o distinto de cero en el caso contrario.

Tasa de muestreo	Valor numérico	Muestras por segundo	Resolución	Resultado mínimo	Resultado máximo
MCP3424_SR_240	0	200	12 bits	0	2047
MCP3424_SR_60	1	60	14 bits	0	8191
MCP3424_SR_15	2	15	16 bits	0	32767
MCP3424_SR_3_75	3	3.75	18 bits	0	131071

Tabla 4: Frecuencias de muestreo disponibles

Ganancia	Valor numérico	Ganancia
MCP3424_GAIN_1	1	x1
MCP3424_GAIN_2	2	x2
MCP3424_GAIN_4	3	x4
MCP3424_GAIN_8	4	x8

Tabla 5: Ganancias disponibles

4.1.1 LECTURA ANÁLOGA: ANALOGREAD ()

int analogRead (int pin);

Realiza una conversión análoga digital sobre el pin *<pin>.*

- *<pin>:* es el número del pin sobre el cual se quiere leer el valor análogo.

Retorna un número entero perteneciente al rango [0,resultado máximo], donde RESULTADO MÁXIMO depende de la tasa de muestreo *<sampleRate>* elegida al inicializar el integrado mediante la función mcp3424Setup(). Ver Tabla 4.

4.2 PWM

La Expander Pi cuenta con 8 pines para realizar PWM en sus proyectos. Para inicializarlos, debe realizar la configuración del integrado PCA9685 llamando a la función pca9685Setup(), en la cual, debe establecer la frecuencia del PWM. Luego podrá establecer el ciclo de trabajo, usando la función pwmWrite().

4.2.1 CONFIGURACIÓN DEL INTEGRADO: PCA9685SETUP()

int pca9685Setup (const int pinBase, const int i2cAddress, int frequency);

Carga la configuración inicial del chip PCA9685, estableciendo el número base de los pines *<pinBase>,* la dirección I²C*<i2cAddress>* y la frecuencia del PWM*<frequency>.*

- *<pinBase>:* número base de los pines de este chip. *<pinBase>*debe ser un número mayor que 64, tal que el rango [pinBase, pinBase + 3] no esté siendo utilizado por otro chip de la librería wiringPi.

<i2cAddress>: dirección del chip MCP3424. La dirección puede ser consultada en la sección 0 "

- ***DIRECCIONES I²C***".
- *<frequency>:* frecuencia del PWM. Debe estar entre el rango [24,1675].Nota: La frecuencia que usted escoja en *<frequency>* puede diferir de la frecuencia entregada por la placa. Esta diferencia aumenta exponencialmente a medida que aumenta *<frequency>*.

  Retorna 0 si no hubo error o distinto de cero en el caso contrario.

5.1.1 ESCRITURA ANALÓGICA (PWM): PWMWRITE()

void pwmWrite (int pin, int value);

Establece el ciclo de trabajo de un pin PWM

- *<pin>:* es el número del pin sobre el cual se configurará su salida PWM
- *<value>:* valor PWM de la salida. Valores entre el rango [0,4095], para ciclos de trabajo menores a 100% (vea Ilustración 2). Valores mayores a 4095 resultarán en un ciclo de trabajo de 100%.

No retorna valor.

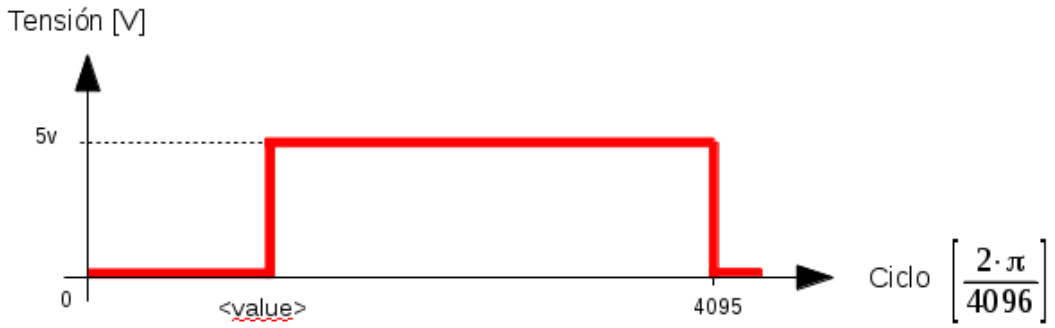


Ilustración 2: Ciclo de trabajo establecido por el valor <value>

EJEMPLOS

En esta sección se presentan códigos de ejemplo para los pines GPIO, ADC y PWM. Compile y ejecute los ejemplos realizando las conexiones que se mencionan. Para terminar la ejecución debe presionar la combinación de teclas *Control+C.*

6.1 ENTRADA GPIO

Conecte un botón con su respectiva resistencia de pull-up o pull-down a +5, GND y al pin 0 del puerto GPIO (ver Ilustración 3). Al ejecutar el programa, verá el estado del pin conectado al botón en pantalla.

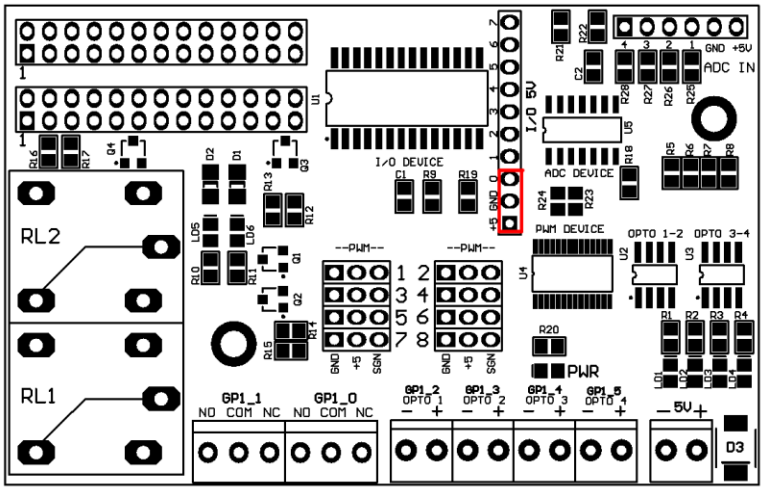


Ilustración 3: Ejemplo 1

//ejemplo expanderPi/ejemplos/inputGPIO.c

```
#include <wiringPi.h>
#include <stdio.h>


#define GPIO_PIN_BASE        200
#define GPIO_DIRECCION_I2C   0x26
#define PIN 0


int main()
{
int buttonState=0;
```

```
wiringPiSetup();
mcp23016Setup(GPIO_PIN_BASE,GPIO_DIRECCION_I2C);
pinMode(GPIO_PIN_BASE + PIN, INPUT);
for(;;)
        {
    buttonState = digitalRead(GPIO_PIN_BASE + PIN);
    printf("buttonState= %d  \r",buttonState);
        fflush(stdout);
        delay(20);
    }
return(0);
}
```

6.2 SALIDA GPIO

Para probar este ejemplo, no es necesario realizar conexión alguna, ya que como salida se utilizará uno de los relés presentes en la placa. Durante la ejecución debe escuchar al relé cambiando de estado, y ver su LED asociado parpadear.

```
//ejemplo expanderPi/ejemplos/outputGPIO.c
#include <wiringPi.h>
#include <stdio.h>

#define GPIO_PIN_BASE         100
#define GPIO_DIRECCION_I2C   0x26
#define PIN 8  //relé

int main()
{
wiringPiSetup();
mcp23016Setup(GPIO_PIN_BASE,GPIO_DIRECCION_I2C);
pinMode(GPIO_PIN_BASE + PIN, OUTPUT);
```

```
for(;;)
        {
        digitalWrite(GPIO_PIN_BASE + PIN, HIGH);
        delay(1000);
        digitalWrite(GPIO_PIN_BASE + PIN, LOW);
        delay(1000);
        }
return(0);
}
```

6.3 ADC

Conecte alguna señal a cualquiera de los pines ADC y la referencia GND (vea la Ilustración 4), y el programa mostrará en pantalla su valor convertido desde análogo a digital.

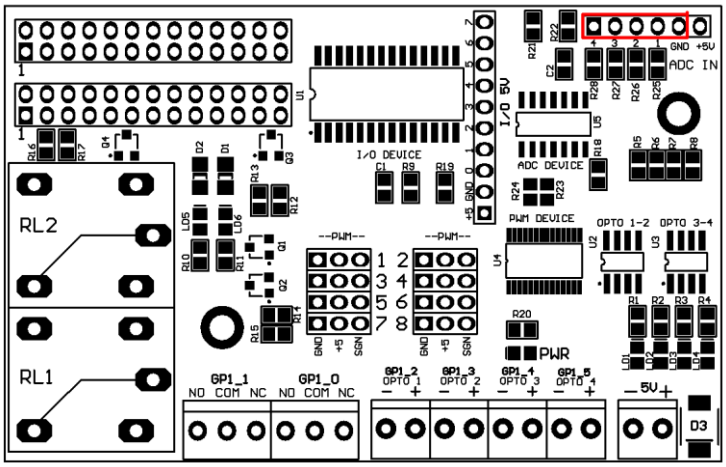


Ilustración 4: Ejemplo 3

```
//ejemplo expanderPi/ejemplos/adc.c
#include <stdio.h>
#include <wiringPi.h>

#define ADC_PIN_BASE        300
#define ADC_DIRECCION_I2C   0x6E
```

```
int main(void)
{
int valor, pin;
wiringPiSetup();
mcp3424Setup(ADC_PIN_BASE, ADC_DIRECCION_I2C,
MCP3424_SR_240, MCP3424_GAIN_1);
for (;;)
                {
        for (pin = 0 ; pin <= 3 ; ++pin)
                {
                valor = analogRead(ADC_PIN_BASE + pin);
                printf("  %5d", valor);
                }
        printf ("\r") ;
        fflush (stdout) ;
        delay (100) ;
        }
return 0 ;
}
```

6.4 PWM

Conecte el cátodo de un LED a GND y el ánodo a una resistencia limitadora, y esta al pin SIGNAL del canal 2 del PWM (vea Ilustración 5). Al correr el programa, verá cambiar la intensidad de brillo del LED.

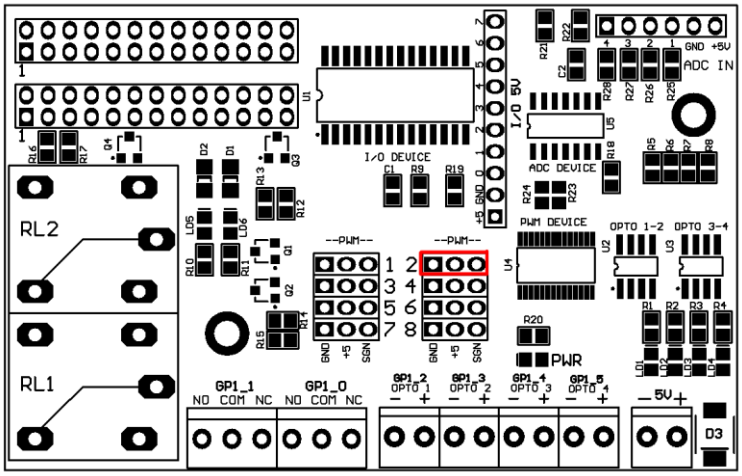


Ilustración 5: Ejemplo 4

```
//ejemplo expanderPi/ejemplos/pwm.c
#include <stdio.h>
#include <wiringPi.h>

#define PWM_PIN_BASE        400
#define PWM_DIRECCION_I2C           0x5C
#define PWM_FRECUENCIA      200

int main ()
{
int i;
wiringPiSetup() ;
pca9685Setup(    PWM_PIN_BASE,   PWM_DIRECCION_I2C,
                        PWM_FRECUENCIA);
for(;;)
        {
        for (i=0 ; i<4096 ; i+=10)
                {
                pwmWrite(PWM_PIN_BASE + 1,i);
```

```
                printf("%5d\r",i);
                fflush(stdout);
                delay(5);
                }
        }
return 0 ;
}
```

SOLUCIÓN DE PROBLEMAS

7.1 LA EXPANDER PI NO REALIZA ALGUNOS COMANDOS QUE LE SON ENVIADOS

Si algunos comandos no son llevados a cabo, el problema puede ser ruido en la comunicación entre las dos placas. Para aumentar la calidad en la transmisión puede intentar disminuir la velocidad (baudrate). Para ello desde la consola ejecute:

gpio load i2c <baudrate>

donde <baudrate> es la velocidad de trasmisión en Kb/s que desea establecer.

Recuerde que al reiniciar se perderá la configuración, por lo que para hacer permanente el cambio, debe establecer dicho valor en el archivo /etc/init.d/expanderPi

7.2 EL EJEMPLO DE SALIDA GPIO CON EL RELÉ FUNCIONA DE FORMA ERRÁTICA.

Si el ejemplo de salida GPIO con el relé presente en la página 16, funciona de forma errática, es decir, a veces escucha el relé y otras no, esto puede deberse a que existe un problema de ruido en la comunicación entre su Raspberry Pi y la Expander Pi. Para solucionarlo, vea la solución del problema frecuente "***La Expander Pi no realiza algunos comandos que le son enviados***".

7.3 ERROR DE COMPILACIÓN: FATAL ERROR: WIRINGPI.H: NO SUCH FILE OR DIRECTORY

El error de compilación "fatal error: wiringPi.h: No such file or directory" ocurre cuando el compilador no puede encontrar el archivo *wiringPi.h*, perteneciente a la librería wiringPi. Para dar solución a este problema, reinstale la librería siguiendo los pasos de la sección 1.1 "***INSTALACIÓN DE LA LIBRERÍA***".

7.4 ERROR DE ENLAZADO: UNDEFINED REFERENCE TO `WIRINGPISETUP'

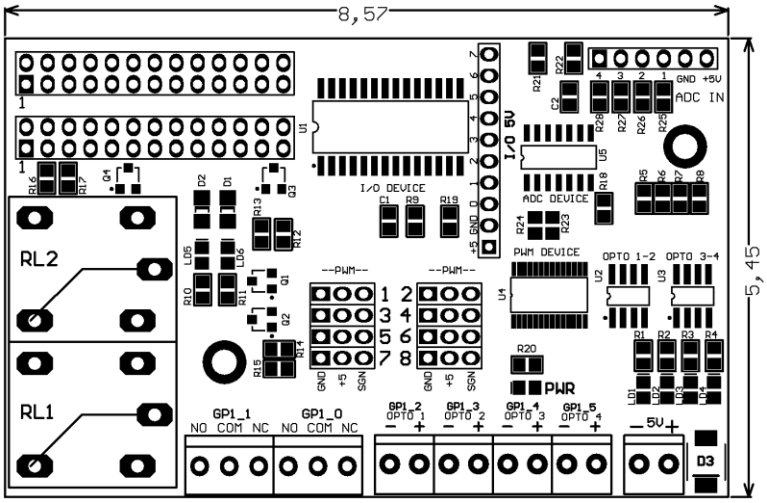
Si al enlazar (paso posterior a la compilación) ocurre el error “undefined reference to `wiringPiSetup'”, tiene un problema con el enlazado de la librería wiringPi. Revise la sección 1.3 “***COMPILACIÓN DE PROGRAMAS***”.

CARACTERÍSTICAS ELÉCTRICAS

- Entrada de voltaje PWM: 5VDC
- Entradas optoacopladas: 24VDC
- Salidas de Relés: 24VDC @1A y 220VAC @1A

CARACTERÍSTICAS MECÁNICAS

Dimensiones: 8,57cm x 5,45cm



HISTORIA DEL DOCUMENTO

Revisión	Fecha	Editado por	Descripción/Cambios
1.0	20 Enero de 2014	Matías Castillo Felmer	Versión inicial del documento